Table of Contents

Python 3 Installation

We will be using Anaconda3 to run Python 3 for this course. Anaconda is a special version of Python that includes a bunch of useful libraries and modules. **This course will expect that everyone is using Anaconda3, so even if you already have Python installed, please follow these instructions! You will still be able to run your old version(s) of Python.**

Complete Python installation as directed below.

For Mac users:

- Open http://continuum.io/downloads with your web browser.
- Download the Python 3 installer for OS X. Double click on what you download.
- Follow the prompts you see in order to install Python 3 Anaconda, using all the defaults for installation.
- After installation is complete, at your Terminal prompt, type: **`python -V`** (note that that should be a CAPITAL V, not a lowercase one! If you make a mistake, it's OK to close the command line and open it again, until you learn nice ways of dealing with that soon!)
- The output you see should look something like this: **`Python 3.5.2 :: Anaconda 4.2.0`**
- At your Terminal prompt, type: **`which python`**.

- The output you see should include the word **`anaconda`**

For Windows users:

If you do not have the Git Bash program installed, do the following:

- Go to **this site**.
- Click on **Download**.
- Scroll toward the bottom of the page. You'll see a link for a Windows executable file that looks like this: **Git-2.11.0-64-bit.exe**. Right click on that link, and then click *Save Link As*. You should see a download progress/appear or however things you save from the Internet normally appear in your browser.
- Double click on that, once it's done downloading. You should see an Installation prompt window. Follow all the installation processes. You can then launch Git Bash the same way you launch any Windows program.

Once you are all set with Git Bash installed:

- Open http://continuum.io/downloads with your web browser.
- Download the Python 3 installer for Windows.
- Install Python 3, with all the default things checked.
    - That should be, *Make Anaconda the default Python* and *Add Anaconda3 to your Path*. But in most cases these should automatically be checked, and you will not need to change anything.
    - Note that this will probably take several minutes!
- Then, open Git Bash, and type exactly the following two lines the prompt (nothing else):
    - **`echo 'export PATH="/c/Anaconda3:$PATH"' > ~/.bashrc`**
    - **`echo 'export PATH="/c/Anaconda3:$PATH"' > ~/.bash_profile`**
    - This command makes it so you will always use the correct version of Python when you run programs in Git Bash. You'll learn more about this later in the semester!
    - Close Git Bash and re-open the Git Bash program.
    - At your Terminal prompt, type: **`python -V`** (note that that should be a CAPITAL V, not a lowercase one! If you make a mistake, it's OK to close the command line and open it again, until you learn nice ways of dealing with that soon!) The output you see should look something like this: **`Python 3.5.2 :: Anaconda 4.2.0`**
    - At your Terminal prompt, type: **`which python`**. The output you see should include the word **`anaconda`**

Making Python 3 your default python

If you have just completed the installation instructions above, Python 3 should be the default python on your system.

If you had installed Python in the previous semester, and are not sure of the Python version, then type this in the terminal/git bash: **`python -V.`** (note that that should be a CAPITAL V)

If it reads **`Python 3.x.x (eg. Python 3.5.2)`**, your default is python version 3, and you need not do anything else.

If it reads **`Python 2.x.x (eg. Python 2.7.10)`**, your default is python version 2. In that case let me know, I will come and help you out :-)

Github Installation

1. Installing Git:

Windows users: Assuming you have Git Bash installed, you are all set!

Mac users: If you do not already have Git / XCode Command Line Tools installed (you probably remember installing them if you did), download Git at this link (click on Mac OS X to download the version for mac): https://git-scm.com/download/mac

2. Getting a GitHub account:

Go to this link: https://github.com/join and create an account on the GitHub website, if you do not already have one.

The username should be something you are willing to share with us and your classmates! We'll use this in class (plus GitHub is public/for sharing, in general).

3. Getting Git ready for use with Github

Open Terminal, and follow the instructions in the links

1. Check for existing SSH Keys: https://help.github.com/articles/checking-for-existing-ssh-keys/#platform-mac

2. If SSH Keys don't exist, create new SSH Keys: https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/#platform-mac

3. Add SSH Keys to your Github.com account: https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/#platform-mac

4. Setup your git user's name and email

```
git config --global user.name "Your Name"
git config --global user.email "email@example.com"
```

GitHub Forking & Cloning

1. Fork the following repository to your Github account: https://github.com/pandeymauli/SI364-DS1
2. You should see <your_user_name>/SI364-DS1 among your Github repositories. Clone that repo to your local machine.
    a. Use Clone with SSH to find the URL. Copy the URL.
    b. In the terminal window, switch to a folder where you would like to clone the repo.
    Hint: use `cd` to switch to different directory
    c. Use git clone <url> to clone the repo. Double check that the URL is similar to git@github.com:<your_username>/ <repo_name>.git
3. Type `ls` to make sure that the repo is correctly cloned to your disk. You should see a folder with name same as the repo's name
4. If you `cd` within the cloned repository and type `ls`, you should see 3 files:
    a. contact-form.html
    b. slides.html
    c. SI364-DS1.py

Designing our own HTML Form

1. We will design our own HTML contact form which includes the following input fields:
    a. Name
    b. Email
    c. Message
    d. Submit button

2. Each input field would have a label describing its purpose.

3. To build our contact form, we will use the following HTML elements:
    a. <label>
    b. <input>
    c. <textarea>
    d. <button>

    You can also play with syntax of these HTML elements here to get practice.

4. We will be editing the file contact-form.html in the cloned repository to prepare our form.

Introduction to Flask App

After completing the form, let us make some edits to a simple flask application. This would be useful in the first homework. In addition, the exercise on forms and flask application will come together in the upcoming lectures.

The application is written in the file SI364-DS1.py. You can open the file in SublimeText to view task instructions. There are 2 tasks. After completing each task, I recommend you push back the changes to GitHub.

Git Cheatsheet

- **`git init`** : Initializing a local git repository on your computer
- **`git add <filename>`** : Staging the files for commit
- **`git commit -m "message"`** : Committing the staged files

- **`git remote add origin <github repository url>`**: Adding a remote repository on GitHub to which you can push to
- **`git push -u origin master`** : Pushing files from local git repo to the remote repo